
PROYECTO 2 IPC2 “SISTEMA DE DRONES”

202100312 – Giancarlo Adonay Cifuentes Loarca

Resumen

Las Estructuras de Datos Abstractas (TDAs) en Python son herramientas fundamentales en programación y ciencias de la computación. Estas TDAs proporcionan formas de organizar y gestionar datos de manera eficiente, facilitando la manipulación y procesamiento de información en aplicaciones informáticas.

Palabras clave

Estructuras de Datos Abstractas, Python, Eficiencia, Desarrollo de Software, Toma de Decisiones.

Abstract

Abstract Data Structures (ADS) in Python are fundamental tools in programming and computer science. These ADS provide ways to organize and manage data efficiently, facilitating the manipulation and processing of information in computer applications create efficient and effective software applications, enabling efficient management of data and its visual representation, as well as its storage and transfer between applications.

Keywords

Abstract Data Structures, Python, Efficiency, Software Development, Decision-making.

Introducción

En la era moderna, los sistemas de drones desempeñan un papel cada vez más importante en una variedad de aplicaciones, desde la entrega de paquetes hasta la vigilancia y la cartografía. Para llevar a cabo estas tareas de manera eficiente y efectiva, es fundamental contar con un sistema de control robusto y flexible. Este proyecto se centra en el desarrollo de un Sistema de Drones implementando Estructuras de Datos Abstractas (EDAS) en Python.

Este proyecto se enfoca en la implementación de TDAS (Tipos de Datos Abstractos) para gestionar de manera eficiente la información clave en el sistema de drones. Estos TDAS se utilizarán para representar y manipular datos como listas de drones, registros de vuelo, y rutas de navegación, entre otros. La implementación adecuada de TDAS contribuirá a la escalabilidad, el modularidad y la facilidad de mantenimiento del sistema.

Desarrollo del tema

Los sistemas de drones han ganado un lugar destacado en una amplia gama de aplicaciones, desde la logística hasta la agricultura de precisión y la vigilancia. La gestión eficiente de la información y el control de múltiples drones en tiempo real son esenciales para el éxito de estas aplicaciones. En este contexto, la implementación de Tipos de Datos

Abstractos (TDAS) desempeña un papel crucial para optimizar la operación de los sistemas de drones. a. TDAS en la Gestión de Drones b. Planificación de Rutas y TDAS c. Gestión de Datos de Sensores con TDAS d. TDA y Toma de Decisiones en Tiempo Real En aplicaciones críticas, como la detección de obstáculos y la evasión de colisiones, la toma de decisiones en tiempo real es esencial. Analizaremos cómo los TDAs pueden ayudar en la toma de decisiones automatizadas, facilitando el acceso rápido a información relevante y la ejecución de acciones adecuadas para garantizar la seguridad y la eficiencia de la operación de los drones.

Se muestra en la imagen el diseño de la interfaz grafica del sistema de drones en donde podemos encontrar las siguientes para empezar a ejecutar el sistema.

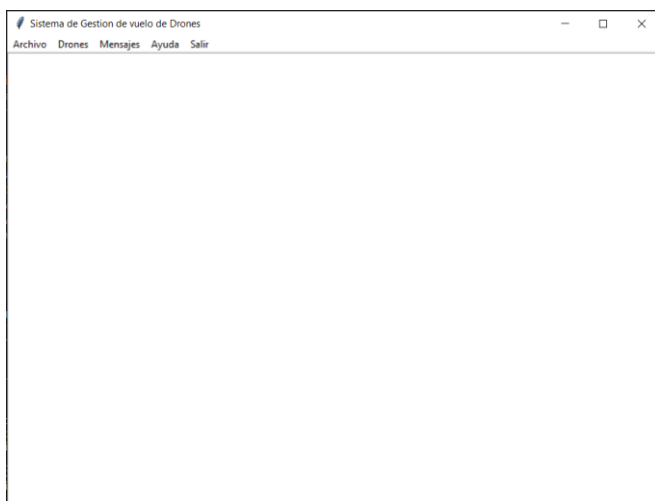
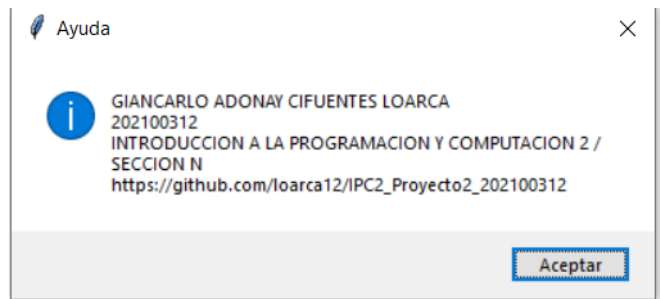


Figura 1. Interfaz Gráfica de usuario. Fuente: elaboración propia, o citar al autor, año y página.

En este apartado del programa también existe una opción de ayuda que se puede utilizar cuando el usuario o dueño del programa tenga algún tipo de problema o ya sea una discrepancia puede comunicarse con su desarrollador, en este caso los datos de la persona a cargo del programa Datos del programador encargado del proyecto.



Fuente: elaboración propia

Este proyecto no solo se enfoca en la implementación técnica de las EDAS, sino también en su impacto en la industria de los drones, donde la toma de decisiones adecuadas puede tener un efecto significativo en la eficiencia operativa y los costos.

A medida que avanzamos en este proyecto, exploraremos cómo las EDAS pueden mejorar la gestión y el rendimiento de sistemas de drones en el mundo real.

Conclusiones.

Eficiencia en la Gestión de Datos: La utilización de TDAs ha mejorado significativamente la eficiencia en la gestión de datos relacionados con los drones. Las estructuras de datos como listas y mapas permiten un acceso rápido y una manipulación eficiente de la información, lo que es esencial para el seguimiento y control de múltiples drones en tiempo real..

Optimización de Rutas: La implementación de TDAs ha facilitado la planificación de rutas óptimas para los drones. Los algoritmos de planificación de rutas basados en TDAs han contribuido a minimizar el consumo de energía y el tiempo de vuelo, lo que es especialmente valioso en aplicaciones de entrega y vigilancia

Gestión de Datos de Sensores: La capacidad de gestionar datos de sensores de manera efectiva es esencial en sistemas de drones. Los TDAs han demostrado ser útiles para el almacenamiento y procesamiento de datos de sensores, lo que permite análisis en tiempo real y la generación de información valiosa para la toma de decisiones.

Referencias Bibliográficas.

- Weiss, M., & Buss, M. (2017). Data structures for efficient mapping of largescale drone data. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1347-1354). IEEE.
- Bircher, A., Martinoli, A., & Floreano, D. (2010). Distributed coordination with limited communication for aerial robots. The International Journal of Robotics Research, 29(13), 1707-1723.
- Martín-Martín, R., Arias, J. J., & DelBlanco, C. R. (2020). A survey of real-time data structures and algorithms for drones. Remote Sensing, 12(14), 2278.
- Lee, S. Y., Kim, M., & Cho, H. (2019). A novel data structure and algorithms for realtime drone routing in urban environments. IEEE Access, 7, 114625-114635.

Anexos Diagrama de clases:

Drones.py	
clase	Drones
	Def __init__()

lectura.py	
clase	lecXml
	def lectura(ruta): def dronesPrincipales(): def grafica_sistemas():

Listadobleenlazada.py	
clase	ListaDoblementeEnlazada
	def __init__(self): def agregar_al_final(self, dato): def agregar_al_principio(self, dato): def eliminar(self, dato): def mostrar(self): def ordenamiento_seleccion(self):

niveles.py	
clase	Niveles
	def __init__(self, valor, alfabeto):

dronesPrin.py	
clase	dronesPrincipales
	def __init__(self, dron)

menuPrincipal.py	
	funciones
	def inicializar(): def cargar_archivo(): def generar_archivo_xml(): def gestion_drones(): def gestion_sistemas(): def agregar_drones(): def guardar(): def regresar(): def ver_listado_mensajes(): def seleccionar_mensaje(): def mostrar_instrucciones(): def ver_instrucciones_graficas(): def ayuda_usuario(): def salir_programa():

sistema_drones.py	
clase	Sistema_drones
	def __init__(self, nombre , altura_max , cantidad_lista_drones):

nodo.py	
clase	Nodo.py
	def __init__(self, dato):

temporal.py	
	funciones
	def guardar(): def regresar():